09/750,317

WEST

Generate Collection Print

L9: Entry 19 of 35 File: USPT Nov 10, 1998

DOCUMENT-IDENTIFIER: US 5835087 A
TITLE: System for generation of object profiles for a system for customized electronic identification of desirable objects

Brief Summary Text (15):
Relevant definitions of terms for the purpose of this description include: (a.) an object available for access by the user, which may be either physical or electronic in nature, is termed a "target object", (b.) a digitally represented profile indicating that target object's attributes is termed a "target profile", (c.) the user looking for the target object is termed a "user", (d.) a profile holding that user's attributes, including age/zip code/etc. is termed a "user profile", (e.) a summary of digital profiles of target objects that a user likes and/or dislikes, is termed the "target profile interest summary" of that user, (f.) a profile consisting of a collection of attributes, such that a user likes target objects whose profiles are similar to this collection of attributes, is termed a "search profile" or in some contexts a "query" or "query profile," (g.) a specific embodiment of the target profile interest summary which comprises a set of search profiles is termed the "search profile set" of a user, (h.) a collection of target objects with similar profiles, is termed a "cluster," (i.) an aggregate profile formed by averaging the attributes of all tar get objects in a cluster, termed a "cluster profile," (j.) a real number determined by calculating the statistical variance of the profiles of all target objects in a cluster, is termed a "cluster variance," (k.) a real number determined by calculating the maximum distance between the profiles of any two target objects in a cluster, is termed a "cluster diameter."

Brief Summary Text (21):
The ability to measure the similarity of profiles describing target objects and a user's interests can be applied in two basic ways: filtering and browsing. Filtering is useful when large numbers of target objects are described in the electronic media s pace. These target objects can for example be articles that are received or potentially received by a user, who only has time to read a small fraction of them. For example, one might potentially receive all items on the AP news wire service, all items posted to a number of news groups, all advertisements in a set of newspapers, or all unsolicited electronic mail, but few people have the time or inclination to read so many articles. A filtering system in the system for customized electronic identification of desirable objects automatically selects a set of articles that the user is likely to wish to read. The accuracy of this filtering system improves over time by noting which articles the user reads and by generating a measurement of the depth to which the user reads each article. This information is then us ed to update the user's target profile interest summary. Browsing provides an alternate method of selecting a small subset of a large number of target objects, such as articles. Articles are organized so that users can actively navigate among groups of articles by moving from one group to a larger, more general group, to a smaller, more specific group, or to a closely related group. Each individual article forms a one-member group of its own, so that the user can navigate to and from individual article s as well as larger groups. The methods used by the system for customized electronic identification of desirable objects allow articles to be grouped into clusters and the clusters to be grouped and merged into larger and larger clusters. These hierarchies of clusters then form the basis for menuing and navigational systems to allow the rapid searching of large numbers of articles. This same clustering technique is applicable to any type of target objects that can be profiled on the electronic media.

Drawing Description Text (5):
FIG. 5 illustrates in flow diagram form a method for automatically generating article profiles and an associated hierarchical menu system;

Drawing Description Text (8):
FIG. 11 illustrates a hierarchical cluster tree example;

Detailed Description Text (109):
Hierarchical clustering of target objects is often useful. Hierarchical clustering produces a tree which divides the target objects first into two large clusters of roughly similar objects; each of these clusters is in turn divided into two or more smaller clusters, which in turn are each divided into yet smaller clusters until the collection of target objects has been entirely divided into "clusters" consisting of a single object each, as diagrammed in FIG. 8 In this diagram, the node d denotes a particular target object d, or equivalently, a single-member cluster consisting of this target object. Target object d is a member of the cluster (a, b, d), which is a subset of the cluster (a, b, c, d, e, f), which in turn is a subset of all target objects. The tree shown in FIG. 8 would be produced from a set of target objects such as those shown geometrically in FIG. 7. In FIG. 7, each letter represents a target object, and axes x1 and x2 represent two of the many numeric attributes on which the target objects differ. Such a cluster tree may be created by hand, using human judgment to form clusters and subclusters of similar objects, or may be created automatically in either of two standard ways: top-down or bottom-up. In top-down hierarchical clustering, the set of all target objects in FIG. 7 would be divided into the clusters (a, b, c, d, e, f) and (g, h, i, j, k). The clustering algorithm would then be reapplied to the target objects in each cluster, so that the cluster (g, h, i, j, k) is subpartitioned into the clusters (g, k) and (h, i, j), and so on to arrive at the tree shown in FIG. 8. In bottom-up hierarchical clustering, the set of all target objects in FIG. 7 would be grouped into numerous small clusters, namely (a, b), d, (c, f), e, (g,k), (h, i), and j. These clusters would then themselves be grouped into the larger clusters (a, b, d), (c, e, f), (g, k), and (b, i, j), according to their cluster profiles. These larger clusters would themselves be grouped into (a, b, c, d, e, f) and (g, k, h, i, j), and so on until all target objects had been grouped together, resulting in the tree of FIG. 8. Note that for bottom-up clustering to work, it must be possible to apply the clustering algorithm to a set of existing clusters. This requires a notion of the distance between two clusters. The method disclosed above for measuring the distance between target objects can be applied directly, provided that clusters are profiled in the same way as target objects. It is only necessary to adopt the convention that a cluster's profile is the average of the target profiles of all the target objects in the cluster; that is, to determine the cluster's value for a given attribute, take the mean value of that attribute across all the target objects in the cluster. For the mean value to be well-defined, all attributes must be numeric, so it is necessary as usual to replace each textual or associative attribute with its decomposition into numeric attributes (scores), as described earlier. For example, the target profile of a single Woody Allen film would assign "Woody-Allen" a score of 1 in the "name-of-director" field, while giving "Federico-Fellini" and "Terence-Davies" scores of 0. A cluster that consisted of 20 films directed by Allen and 5 directed by Fellini would be profiled with scores of 0.8, 0.2, and 0 respectively, because, for example, 0.8 is the average of 20 ones and 5 zeros.

Detailed Description Text (111):
Given a target object with target profile P, or alternatively given a search profile P, a hierarchical cluster tree of target objects makes it possible for the system to search efficiently for target objects with target profiles similar to P. It is only necessarily to navigate through the tree, automatically, in search of such target profiles. The system for customized electronic identification of desirable objects begins by considering the largest, top-level clusters, and selects the cluster whose profile is most similar to target profile P. In the event of a near-tie, multiple clusters may be selected. Next, the system considers all subclusters of the selected clusters, and this time selects the subcluster or subclusters whose profiles are closest to target profile P. This refinement process is iterated until the clusters selected on a given step are sufficiently small, and these are the desired clusters of target objects with profiles most similar to target profile P. Any hierarchical cluster tree therefore serves as a decision tree for identifying target objects. In pseudo-code form, this process is as follows (and in flow diagram form in FIGS. 13A and 13B):

Detailed Description Text (113):
2. Initialize the current tree T to be the hierarchical cluster tree of all objects at step 13A01 and at step 13A02 scan the current cluster tree for target objects similar to P, using the process detailed in FIG. 13B. At step 13A03, the list of target objects is returned.

Detailed Description Text (119):

In step 5 of this pseudo-code, smaller thresholds are typically used at lower levels of the tree, for example by making the threshold an affine function or other function of the cluster variance or cluster diameter of the cluster pi. If the cluster tree is distributed across a plurality of servers, as described in the section of this description titled "Network Context of the Browsing System", this process may be executed in distributed fashion as follows: steps 3-7 are executed by the server that stores the root node of hierarchical cluster tree T, and the recursion in step 7 to a subcluster tree T.sub.i involves the transmission of a search request to the server that stores the root node of tree T.sub.i, which server carries out the recursive step upon receipt of this request. Steps 1-2 are carried out by the processor that initiates the search, and the server that executes step 6 must send a message identifying the target object to this initiating processor, which adds it to the list.

Detailed Description Text (120):
Assuming that low-level clusters have been already been formed through clustering, there are alternative search methods for identifying the low-level cluster whose profile is most similar to a given target profile P. A standard back-propagation neural net is one such method: it should be trained to take the attributes of a target object as input, and produce as output a unique pattern that can be used to identify the appropriate low-level cluster. For maximum accuracy, low-level clusters that are similar to each other (close together in the cluster tree) should be given similar identifying patterns. Another approach is a standard decision tree that considers the attributes of target profile P one at a time until it can identify the appropriate cluster. If profiles are large, this may be more rapid than considering all attributes. A hybrid approach to searching uses distance measurements as described above to navigate through the top few levels of the hierarchical cluster tree, until it reaches an cluster of intermediate size whose profile is similar to target profile P, and then continues by using a decision tree specialized to search for low-level subclusters of that intermediate cluster.

Detailed Description Text (121):
One use of these searching techniques is to search for target objects that match a search profile from a user's search profile set. This form of searching is used repeatedly in the news clipping service, active navigation, and Virtual Community Service applications, described below. Another use is to add a new target object quickly to the cluster tree. An existing cluster that is similar to the new target object can be located rapidly, and the new target object can be added to this cluster. If the object is beyond a certain threshold distance from the cluster center, then it is advisable to start a new cluster. Several variants of this incremental clustering scheme can be used, and can be built using variants of subroutines available in advanced statistical packages. Note that various methods can be used to locate the new target objects that must be added to the cluster tree, depending on the architecture used. In one method, a "webcrawler" program running on a central computer periodically scans all servers in search of new target objects, calculates the target profiles of these objects, and adds them to the hierarchical cluster tree by the above method. In another, whenever a new target object is added to any of the servers, a software "agent" at that server calculates the target profile and adds it to the hierarchical cluster tree by the above method.

Detailed Description Text (127):
Similar information can alternatively be extracted from a collection of consumer profiles without recourse to a decision tree, by considering attributes one at a time, and identifying those attributes on which product X's consumers differ significantly from its non-consumers. These techniques serve to characterize consumers of a particular product; they can be equally well applied to voter research or other survey research, where the objective is to characterize those individuals from a given set of surveyed individuals who favor a particular candidate, hold a particular opinion, belong to a particular demographic group, or have some other set of distinguishing attributes. Researchers may wish to purchase batches of analyzed or unanalyzed user profiles from which personal identifying information has been removed. As with any statistical database, statistical conclusions can be drawn, and relationships between attributes can be elucidated using knowledge discovery techniques which are well known in the art.

Detailed Description Text (161):
In other scenarios, the request R to proxy server S2 formed by the user may have different content. For example, request R may instruct proxy server S2 to use the methods described later in this description to retrieve from the most convenient server a particular piece of information that has been multicast to many servers, and to send

this information to the user. Conversely, request R may instruct proxy server S2 to multicast to many servers a file associated with a new target object provided by the user, as described below. If the user is a subscriber to the news clipping service described below, request R may instruct proxy server S2 to forward to the user all target objects that the news clipping service has sent to proxy server S2 for the user's attention. If the user is employing the active navigation service described below, request R may instruct proxy server S2 to select a particular cluster from the hierarchical cluster tree and provide a menu of its subclusters to the user, or to activate a query that temporarily affects proxy server S2's record of the user's target profile interest summary. If the user is a member of a virtual community as described below, request R may instruct proxy server S2 to forward to the user all messages that have been sent to the virtual community.

Detailed Description Text (176):
Pre-fetching of locally stored data has been heavily studied in memory hierarchies, including CPU caches and secondary storage (disks), for several decades. A leader in this area has been A. J. Smith of Berkeley, who identified a variety of schemes and analyzed opportunities using extensive traces in both databases and CPU caches. His conclusion was that general schemes only really paid off where there was some reasonable chance that sequential access was occurring, e.g, in a sequential read of data. As the balances between various latencies in the memory hierarchy shifted during the late 1980's and early 1990's, J. M. Smith and others identified further opportunities for pre-fetching of both locally stored data and network data. In particular, deeper analysis of patterns in work by Blaha showed the possibility of using expert systems for deep pattern analysis that could be used for pre-fetching. Work by J. M. Smith proposed the use of reference history trees to anticipate references in storage hierarchies where there was some historical data. Recent work by Touch and the Berkeley work addressed the case of data on the World-Wide Web, where the large size of images and the long latencies provide extra incentive to pre-fetch; Touch's technique is to pre-send when large bandwidths permit some speculation using HTML storage references embedded in WEB pages, and the Berkeley work uses techniques similar to J. M. Smith's reference histories specialized to the semantics of HTML data.

Detailed Description Text (177):
Successful pre-fetching depends on the ability of the system to predict the next action or actions of the user. In the context of the system for customized electronic identification of desirable objects, it is possible to cluster users into groups according to the similarity of their user profiles. Any of the well-known pre-fetching methods that collect and utilize aggregate statistics on past user behavior, in order to predict future user behavior, may then be implemented in so as to collect and utilize a separate set of statistics for each cluster of users. In this way, the system generalizes its access pattern statistics from each user to similar users, without generalizing among users who have substantially different interests. The system may further collect and utilize a similar set of statistics that describes the aggregate behavior of all users; in cases where the system cannot confidently make a prediction as to what a particular user will do, because the relevant statistics concerning that user's user cluster are derived from only a small amount of data, the system may instead make its predictions based on the aggregate statistics for all users, which are derived from a larger amount of data. For the sake of concreteness, we now describe a particular instantiation of a pre-fetching system, that both employs these insights and that makes its pre-fetching decisions through accurate measurement of the expected cost and benefit of each potential pre-fetch.

Detailed Description Text (219):
(c.) satisfying the request would involve disclosure to the accessor of a certain fact about the user's user profile

Detailed Description Text (220):
(d.) satisfying the request would involve disclosure to the accessor of the user's target profile interest summary

Detailed Description Text (221):
(e.) satisfying the request would involve disclosure to the accessor of statistical summary data, which data are computed from the user's user profile or target profile interest summary together with the user profiles and target profile interest summaries of at least n other users in the user base of the proxy server

Detailed Description Text (249):

A set of topical multicast trees for a set of homogenous target objects may be constructed or reconstructed at any time, as follows. The set of target objects is grouped into a fixed number of topical clusters C1 . . . Cp with the methods described above, for example, by choosing C1 . . . Cp to be the result of a k-means clustering of the set of target objects, or alternatively a covering set of low-level clusters from a hierarchical cluster tree of these target objects. A multicast tree MT(c) is then constructed from each cluster C in C1 . . . Cp, by the following procedure:

Detailed Description Text (283):
These news articles are then hierarchically clustered in a hierarchical cluster tree at step 503, which serves as a decision tree for determining which news articles are closest to the user's interest. The resulting clusters can be viewed as a tree in which the top of the tree includes all target objects and branches further down the tree represent divisions of the set of target objects into successively smaller subclusters of target objects. Each cluster has a cluster profile, so that at each node of the tree, the average target profile (centroid) of all target objects stored in the subtree rooted at that node is stored. This average of target profiles is computed over the representation of target profiles as vectors of numeric attributes, as described above.

Detailed Description Text (285):
The process by which a user employs this apparatus to retrieve news articles of interest is illustrated in flow diagram form in FIG. 11. At step 1101, the user logs into the data communication network N via their client processor C.sub.1 and activates the news reading program. This is accomplished by the user establishing a pseudonymous data communications connection as described above to a proxy server S.sub.2, which provides front-end access to the data communication network N. The proxy server S.sub.2 maintains a list of authorized pseudonyms and their corresponding public keys and provides access and billing control. The user has a search profile set stored in the local data storage medium on the proxy server S.sub.2. When the user requests access to "news" at step 1102, the profile matching module 203 resident on proxy server S.sub.2 sequentially considers each search profile P.sub.k from the user's search profile set to determine which news articles are most likely of interest to the user. The news articles were automatically clustered into a hierarchical cluster tree at an earlier step so that the determination can be made rapidly for each user. The hierarchical cluster tree serves as a decision tree for determining which articles' target profiles are most similar to search profile P.sub.k : the search for relevant articles begins at the top of the tree, and at each level of the tree the branch or branches are selected which have cluster profiles closest to p.sub.k. This process is recursively executed until the leaves of the tree are reached, identifying individual articles of interest to the user, as described in the section "Searching for Target Objects" above.

Detailed Description Text (286):
A variation on this process exploits the fact that many users have similar interests. Rather than carry out steps 5-9 of the above process separately for each search profile of each user, it is possible to achieve added efficiency by carrying out these steps only once for each group of similar search profiles, thereby satisfying many users' needs at once. In this variation, the system begins by non-hierarchically clustering all the search profiles in the search profile sets of a large number of users. For each cluster k of search profiles, with cluster profile P.sub.k, it uses the method described in the section "Searching for Target Objects" to locate articles with target profiles similar to P.sub.k. Each located article is then identified as of interest to each user who has a search profile represented in cluster k of search profiles.

Detailed Description Text (287):
Notice that the above variation attempts to match clusters of search profiles with similar clusters of articles. Since this is a symmetrical problem, it may instead be given a symmetrical solution, as the following more general variation shows. At some point before the matching process commences, all the news articles to be considered are clustered into a hierarchical tree, termed the "target profile cluster tree," and the search profiles of all users to be considered are clustered into a second hierarchical tree, termed the "search profile cluster tree." The following steps serve to find all matches between individual target profiles from any target profile cluster tree and individual search profiles from any search profile cluster tree: 1. For each child subtree S of the root of the search profile cluster tree (or, let S be the entire search profile cluster tree if it contains only one search profile): 2. Compute the cluster profile P.sub.S to be the average of all search profiles in subtree S 3. For each subcluster (child subtree) T of the root of the target profile cluster tree (or, let T be the entire target profile cluster tree if it contains only one target

profile): 4. Compute the cluster profile P.sub.T to be the average of all target profiles in subtree T 5. Calculate d(P.sub.S, P.sub.T; the distance between P.sub.S and P.sub.T 6. If d(P.sub.S, P.sub.T)<t, a threshold, 7. If S contains only one search profile and T contains only one target profile, decl are a match between that search profile and that target profile, 8. otherwise recurse to step 1 to find all matches between search profiles in tree S and target profiles in tree T.

Detailed Description Text (323):
A hierarchical cluster tree imposes a useful organization on a collection of target objects. The tree is of direct use to a user who wishes to browse through all the target objects in the tree. Such a user may be exploring the collection with or without a well-specified goal. The tree's division of target objects into coherent clusters provides an efficient method whereby the user can locate a target object of interest. The user first chooses one of the highest level (largest) clusters from a menu, and is presented with a menu listing the subclusters of said cluster, whereupon the user may select one of these subclusters. The system locates the subcluster, via the appropriate pointer that was stored with the larger cluster, and allows the user to select one of its subclusters from another menu. This process is repeated until the user comes t o a leaf of the tree, which yields the details of an actual target object. Hierarchical trees allow rapid selection of one target object from a large set. In ten menu selections from menus of ten items (subclusters) each, one can reach 10.sup.10 =10,000,000,000 (ten billion) items. In the preferred embodiment, the user views the menus on a computer screen or terminal screen and selects from them with a keyboard or mouse. However, the user may also make selections over the telephone, with a voice synthesizer reading the menus and the user selecting subclusters via the telephone's touch-tone keypad. In another variation, the user simultaneously maintains two connections to the server, a telephone voice connection and a fax connection; the server sends successive menus to the user by fax, while the user selects choices via the telephone's touch-tone keypad.

Detailed Description Text (324):
Just as user profiles commonly include an associative attribute indicating the user's degree of interest in each target object, it is useful to augment user profiles with an additional associative attribute indicating the user's degree of interest in each cluster in the hierarchical cluster tree. This degree of interest may be estimated numerically as the number of subclusters or target objects the user has selected from menus associated with the given cluster or its subclusters, expressed as a proportion of the total number of subclusters or target objects the user has selected. This associative attribute is particularly valuable if the hierarchical tree was built using "soft" or "fuzzy" clustering, which allows a subcluster or target object to appear in multiple clusters: if a target document appears in both the "sports" and the "humor" clusters, and the user selects it from a menu associated with the "humor" cluster, then the system increases its association between the user and the "humor" cluster but not its association between the user and the "sports" cluster.

Detailed Description Text (330):
It should be appreciated that a hierarchical cluster-tree may be configured with multiple cluster selections branching from each node or the same labeled clusters presented in the form of single branches for multiple nodes ordered in a hierarchy. In one variation, the user is able to perform lateral navigation between neighboring clusters as well, by requesting that the system search for a cluster whose cluster profile resembles the cluster profile of the currently selected cluster. If this type of navigation is performed at the level of individual objects (leaf ends), then automatic hyperlinks may be then created as navigation occurs. This is one way that nearest neighbor clustering navigation may be performed. For example, in a domain where target objects are home pages on the World Wide Web, a collection of such pages could be laterally linked to create a "virtual mall."

Detailed Description Text (336):
Although the topology of a hierarchical cluster tree is fixed by the techniques that build the tree, the hierarchical menu presented to the user for the user's navigation need not be exactly isomorphic to the cluster tree. The menu is typically a somewhat modified version of the cluster tree, reorganized manually or automatically so that the clusters most interesting to a user are easily accessible by the user. In order to automatically reorganize the menu in a user-specific way, the system first attempts automatically to identify existing clusters that are of interest to the user. The system may identify a cluster as interesting because the user often accesses target objects in that cluster--or, in a more sophisticated variation, because the user is predicted to have high interest in the cluster's profile, using the methods disclosed

herein for estimating interest from relevance feedback.

Detailed Description Text (337):
Several techniques can then be used to make interesting clusters more easily accessible. The system can at the user's request or at all times display a special list of the most interesting clusters, or the most interesting subclusters of the current cluster, so that the user can select one of these clusters based on its label and jump directly to it. In general, when the system constructs a list of interesting clusters in this way, the I.sup.th most prominent choice on the list, which choice is denoted Top(), is found by considering all appropriate clusters C that are further than a threshold distance t from all of Top(1), Top(2), . . . Top(I-1), and selecting the one in which the user's interest is estimated to be highest. Here the threshold distance t is optionally dependent on the computed cluster variance or cluster diameter of the profiles in the latter cluster. Several techniques that reorganize the hierarchical menu tree are also useful. First, menus can be reorganized so that the most interesting subcluster choices appear earliest on the menu, or are visually marked as interesting; for example, their labels are displayed in a special color or type face, or are displayed together with a number or graphical image indicating the likely level of interest. Second, interesting clusters can be moved to menus higher in the tree, i.e., closer to the root of the tree, so that they are easier to access if th e user starts browsing at the root of the tree. Third, uninteresting clusters can be moved to menus lower in the tree, to make room for interesting clusters that are being moved higher. Fourth, clusters with an especially low interest score (representing active dislike) can simply be suppressed from the menus; thus, a user with children may assign an extremely negative weight to the "vulgarity" attribute in the determination of q, so that vulgar clusters and documents will not be available at all. As the interesting clusters and the documents in them migrate toward the top of the tree, a customized tree develops that can be more efficiently navigated by the particular user. If menus are chosen so that each menu item is chosen with approximately equal probability, then the expected number of choices the user has to make is minimized. If, for example, a user frequently accessed target objects whose profiles resembled the cluster profile of cluster (a, b, d) in FIG. 8 then the menu in FIG. 9 could be modified to show the structure illustrated in FIG. 10.

Detailed Description Text (339):
In a system where queries are used, it is useful to include in the target profiles an associative attribute that records the associations between a target object and whatever terms are employed in queries used to find that target object. The association score of target object X with a particular query term T is defined to be the mean relevance feedback on target object X, averaged over just those accesses of target object X that were made while a query containing term T was active, multiplied by the negated logarithm of term T's global frequency in all queries. The effect of this associative attribute is to increase the measured similarity of two documents if they are good responses to queries that contain the same terms. A further maneuver can be used to improve the accuracy of responses to a query: in the summation used to determine the quality q(TJ, X) of a target object X, a term is included that is proportional to the sum of association scores between target object X and each term in the active query, if any, so that target objects that are closely associated with terms in an active query are determined to have higher quality and therefore higher interest for the user. To complement the system's automatic reorganization of the hierarchical cluster tree, the user can be given the ability to reorganize the tree manually, as he or she sees fit. Any changes are optionally saved on the user's local storage device so that they will affect the presentation of the tree in future sessions. For example, the user can choose to move or copy menu options to other menus, so that useful clusters can thereafter be chosen directly from the root menu of the tree or from other easily accessed or topically appropriate menus. In an other example, the user can select clusters C.sub.1, C.sub.2, . . . C.sub.k listed on a particular menu M and choose to remove these clusters from the menu, replacing them on the menu with a single aggregate cluster M' containing all the target objects from clusters C.sub.1, C.sub.2, . . . C.sub.k. In this case, the immediate subclusters of new cluster M' are either taken to be clusters C.sub.1, C.sub.2, ... Ck themselves, or else, in a variation similar to the "scatter-gather" method, are automatically computed by clustering the set of all the subclusters of clusters C.sub.1, C.sub.2, . . . C.sub.k according to the similarity of the cluster profiles of these subclusters.

Detailed Description Text (341):
In one application, the browsing techniques described above may be applied to a domain where the target objects are purchasable goods. When shoppers look for goods to purchase over the Internet or other electronic media, it is typically necessary to

display thousands or tens of thousands of products in a fashion that helps consumers find the items they are looking for. The current practice is to use hand-crafted menus and sub-menus in which similar items are grouped together. It is possible to use the automated clustering and browsing methods described above to more effectively group and present the items. Purchasable items can be hierarchically clustered using a plurality of different criteria. Useful attributes for a purchasable item include but are not limited to a textual description and predefined category labels (if available), the unit price of the item, and an associative attribute listing the users who have bought this item in the past. Also useful is an associative attribute indicating which other items are often bought on the same shopping "trip" as this item; items that are often bought on the same trip will be judged similar with respect to this attribute, so tend to be grouped together. Retailers may be interested in utilizing a similar technique for purposes of predicting both the nature and relative quantity of items which are likely to be popular to their particular clientele. This prediction may be made by using aggregate purchasing records as the search profile set from which a collection of target objects is recommended. Estimated customer demand which is indicative of (relative) inventory quantity for each target object item is determined by measuring the cluster variance of that item compared to another target object item (which is in stock).

Detailed Description Text (342):
As described above, hierarchically clustering the purchasable target objects results in a hierarchical menu system, in which the target objects or clusters of target objects that appear on each menu can be labeled by names or icons and displayed in a two-dimensional or three-dimensional menu in which similar items are displayed physically near each other or on the same graphically represented "shelf." As described above, this grouping occurs both at the level of specific items (such as standard size Ivory soap or large Breck shampoo) and at the level of classes of items (such as soaps and shampoos). When the user selects a class of items (for instance, by clicking on it), then the more specific level of detail is displayed. It is neither necessary nor desirable to limit each item to appearing in one group; customers are more likely to find an object if it is in multiple categories. Non-purchasable objects such as artwork, advertisements, and free samples may also be added to a display of purchasable objects, if they are associated with (liked by) substantially the same users as are the purchasable objects in the display.

Detailed Description Text (344):
The files associated with target objects are typically distributed across a large number of different servers S1-So and clients C1-Cn. Each file has been entered into the data storage medium at some server or client in any one of a number of ways, including, but not limited to: scanning, keyboard input, e-mail, FTP transmission, automatic synthesis from another file under the control of another computer program. While a system to enable users to efficiently locate target objects may store its hierarchical cluster tree on a single centralized machine, greater efficiency can be achieved if the storage of the hierarchical cluster tree is distributed across many machines in the network. Each cluster C, including single-member clusters (target objects), is digitally represented by a file F, which is multicast to a topical multicast tree MT(C1); here cluster C1 is either cluster C itself or some supercluster of cluster C. In this way, file F is stored at multiple servers, for redundancy. The file F that represents cluster C contains at least the following data:

Detailed Description Text (348):
The distributed hierarchical cluster tree can be created in a distributed fashion, that is, with the participation of many processors. Indeed, in most applications it should be recreated from time to time, because as users interact with target objects, the associative attributes in the target profiles of the target objects change to reflect these interactions; the system's similarity measurements can therefore take these interactions into account when judging similarity, which allows a more perspicuous cluster tree to be built The key technique is the following procedure for merging n disjoint cluster trees, represented respectively by files F1 . . . Fn in distributed fashion as described above, into a combined cluster tree that contains all the target objects from all these trees. The files F1 . . . Fn are described above, except that the cluster labels are not included in the representation. The following steps are executed by a server S1, in response to a request message from another server S0, which request message includes pointers to the files F1 . . . Fn. 1. Retrieve files F1 . . . Fn. 2. Let L and M be empty lists. 3. For each file Fi from among F1 . . . Fn: 4. If file Fi contains pointers to subcluster files, add these pointers to list L. 5. If file Fi represents a single target object, add a pointer to file Fi to list L. 6. For each pointer X on list L, retrieve the file that pointer P points to and extract the cluster

profile P(X) that this file stores. 7. Apply a clustering algorithm to group the pointers X on list L according to the distances between their respective cluster profiles P (X). 8. For each (nonempty) resulting group C of pointers: 9. If C contains only one pointer, add this pointer to list M; 10. otherwise, if C contains exactly the same subcluster pointers as does one of the files Fi from among F1 . . . Fn, then add a pointer to file Fi to list M; 11. otherwise: 12. Select an arbitrary server S2 on the network, for example by randomly selecting one of the pointers in group C and choosing the server it points to. 13. Send a request message to server S2 that includes the subcluster pointers in group C and requests server S2 to merge the corresponding subcluster trees. 14. Receive a response from server S2, containing a pointer to a file G that represents the merged tree. Add this pointer to list M. 15. For each file Fi from among F1 . . . Fn: 16. If list M does not include a pointer to file Fi, send a message to the server or servers storing Fi instructing them to delete file Fi. 17. Create and store a file F that represents a new cluster, whose subcluster pointers are exactly the subcluster pointers on list M. 18. Send a reply message to server S0, which reply message contains a pointer to file F and indicates that file F represents the merged cluster tree.

Detailed Description Text (349):
With the help of the above procedure, and the multicast tree MT full that includes all proxy servers in the network, the distributed hierarchical cluster tree for a particular domain of target objects is constructed by merging many local hierarchical cluster trees, as follows. 1. One server S (preferably one with good connectivity) is elected from the tree. 2. Server S sends itself a global request message that causes each proxy server in MT.sub.full (that is, each proxy server in the network) to ask its clients for files for the cluster tree. 3. The clients of each proxy server transmit to the proxy server any files that they maintain, which files represent target objects from the appropriate domain that should be added to the cluster tree. 4. Server S forms a request R1 that, upon receipt, will cause the recipient server S1 to take the following actions: (a) Build a hierarchical cluster tree of all the files stored on server S1 that are maintained by users in the user base of S1. These files correspond to target objects from the appropriate domain. This cluster tree is typically stored entirely on S1, but may in principle be stored in a distributed fashion. (b) Wait until all servers to which the server S1 has propagated request R have sent the recipient reply messages containing pointers to cluster trees. (c) Merge together the cluster tree created in step 5(a) and the cluster trees supplied in step 5(b), by sending any server (such as S1 itself) a message requesting such a merge, as described above. (d) Upon receiving a reply to the message sent in (c), which reply includes a pointer to a file representing the merged cluster tree, forward this reply to the sender of request R1, unless this is S1 itself 5. Server S sends itself a global request message that causes all servers in MT.sub.full to act on embedded request R1. 6. Server S receives a reply to the message it sent in 5(c). This reply includes a pointer to a file F that represents the completed hierarchical cluster tree. Server S multicasts file F to all proxy servers in MT.sub.full. Once the hierarchical cluster tree has been created as above, server S can send additional messages through the cluster tree, to arrange that multicast trees MT(C) are created for sufficiently large clusters C, and that each file F is multicast to the tree MT(C), where C is the smallest cluster containing file F.

Detailed Description Text (352):
Computer users frequently join other users for discussions on computer bulletin boards, newsgroups, mailing lists, and real-time chat sessions over the computer network, which may be typed (as with Internet Relay Chat (IRC)), spoken (as with Internet phone), or videoconferenced. These forums are herein termed "virtual communities." In current practice, each virtual community has a specified topic, and users discover communities of interest by word of mouth or by examining a long list of communities (typically hundreds or thousands). The users then must decide for themselves which of thousands of messages they find interesting from among those posted to the selected virtual communities, that is, made publicly available to members of those communities. If they desire, they may also write additional messages and post them to the virtual communities of their choice. The existence of thousands of Internet bulletin boards (also termed newsgroups) and countless more Internet mailing lists and private bulletin board services (BBS's) demonstrates the very strong interest among members of the electronic community in forums for the discussion of ideas about almost any subject imaginable. Presently, virtual community creation proceeds in a haphazard form, usually instigated by a single individual who decides that a topic is worthy of discussion. There are protocols on the Internet for voting to determine whether a newsgroup should be created, but there is a large hierarchy of newsgroups (which begin with the prefix "alt.") that do not follow this protocol.

Detailed Description Text (373):
A separate multicast tree MT(V) is maintained for each virtual community V, by use of the following four procedures. 1. To construct or reconstruct this multicast tree, the core servers for virtual community V are taken to be those proxy servers that serve at least one pseudonymous member of virtual community V. Then the multicast tree MT(V) is established via steps 4-6 in the section "Multicast Tree Construction Procedure" above. 2. When a new user joins virtual community V, which is an existing virtual community, the user sends a message to the user's proxy server S. If user's proxy server S is not already a core server for V, then it is designated as a core server and is added to the multicast tree MT(V), as follows. If more than k servers have been added since the last time the multicast tree MT(V) was rebuilt, where k is a function of the number of core servers already in the tree, then the entire tree is simply rebuilt via steps 4-6 in the section "Multicast Tree Construction Procedure" above. Otherwise, server S retrieves its locally stored list of nearby core servers for V, and chooses a server S1. Server S sends a control message to S1, indicating that it would like to be added to the multicast tree MT(V). Upon receipt of this message, server SI retrieves its locally stored subtree G1 of MT(V), and forms a new graph G from G1 by removing all degree-1 vertices other than S1 itself. Server S1 transmits graph G to server S, which stores it as its locally stored subtree of MT(V). Finally, server S sends a message to itself and to all servers that are vertices of graph G, instructing these servers to modify their locally stored subtrees of MT(V) by adding S as a vertex and adding an edge between SI and S. 3. When a user at a client q wishes to send a message F to virtual community V, client q embeds message F in a request R instructing the recipient to store message F locally, for a limited time, for access by member s of virtual community V. Request R includes a credential proving that the user is a member of virtual community V or is otherwise entitled to post messages to virtual community V (for example is not "black marked" by that or other virtual community members). Client q then broadcasts request R to all core servers in the multicast tree MT(V), by means of a global request message transmitted to the user's proxy server as described above. The core servers satisfy request R, provided that they can verify the included credential. 4. In order to retrieve a particular message sent to virtual community V, a user U at client q initiates the steps described in section "Retrieving Files from a Multicast Tree," above. If user U does not want to retrieve a particular message, but rather wants to retrieve all new messages sent to virtual community V, then user U pseudonymously instructs its proxy server (which is a core server for V) to send it all messages that were multicast to MT(V) after a certain date. In either case, user U must provide a credential proving user U to be a member of virtual community V, or otherwise entitled to access messages on virtual community V.

Other Reference Publication (20):
Willett, P., "Recent Trends in Hierarchic Document Clustering: A Critical Review", Information Processing & Management, vol. 24, No. 5, pp. 557-597, 1988.

CLAIMS:

1. A method for cataloging a plurality of target objects that are stored on an electronic storage media, where users are connected via user terminals and bidirectional data communication connections to a target server that accesses said electronic storage media, said method comprising the steps of:

storing on said electronic storage media each target object;

automatically generating in said target server, target profiles for each of said target objects that are stored on said electronic storage media, each of said target profiles being generated from the contents of an associated one of said target objects and their associated target object characteristics comprising:

automatically generating a hierarchical menu that directs said users to at least a subset of said plurality of target objects stored on said electronic media, comprising:

sorting all target objects in said subset into a plurality of clusters of target objects based on an empirical measure of similarity of content of said target objects, and

generating a hierarchical menu that identifies the content in common of target objects sorted into each of said plurality of clusters, to enable said identified user to identify ones of said plurality of target objects stored on said electronic storage media that are likely to be of interest to said identified user.

2. The method of claim 1 wherein said step of automatically generating a hierarchical menu further comprises:

ascribing a cluster profile to each of said plurality of clusters.

3. The method of claim 1 wherein said step of sorting comprises:

dividing said plurality of target objects into at least two clusters based upon said empirical measure of similarity of content of said target objects;

subdividing each of said at least two clusters into at least two subclusters based upon said empirical measure of similarity of content of said target objects contained in each said cluster; and

repeating said step of subdividing to produce a multi-level hierarchy of identified clusters.

4. The method of claim 3 wherein said step of generating a hierarchical menu comprises:

ascribing a cluster profile to each cluster produced by all steps of dividing and subdividing in said step of sorting.

7. A method for cataloging a plurality of target objects that are stored on an electronic storage media, where users are connected via user terminals and bidirectional data communication connections to a target server that accesses said electronic storage media, and wherein said sets of target object characteristics comprise sets of user interest characteristics for a virtual community of users, said method comprising the steps of:

storing on said electronic storage media each target object;

automatically generating in said target server, target profiles for each of said target objects that are stored on said electronic storage media, each of said target profiles being generated from the contents of an associated one of said target objects and their associated target object characteristics comprising:

automatically generating a hierarchical menu that directs a requesting one of said users to at least a subset of said sets of user interest characteristics for a virtual community of users stored on said electronic media, comprising:

sorting all sets of user interest characteristics for a virtual community of users in said subset into a plurality of clusters of target sets of user interest characteristics for a virtual community of users based on an empirical measure of similarity of content of said target sets of user interest characteristics for a virtual community of users, and

generating a hierarchical menu that identifies the content in common of target sets of user interest characteristics for a virtual community of users sorted into each of said plurality of clusters, to enable said requesting user to identify ones of said plurality of target sets of user interest characteristics for a virtual community of users stored on said electronic storage media that are likely to be of interest to said requesting user.

8. The method of claim 7 wherein said step of automatically generating a hierarchical menu further comprises:

ascribing a cluster profile to each of said plurality of clusters.

9. The method of claim 7 wherein said step of sorting comprises:

dividing said plurality of target sets of user interest characteristics for a virtual community of users into at least two clusters based upon said empirical measure of similarity of content of said target sets of user interest characteristics for a virtual community of users;

subdividing each of said at least two clusters into at least two subclusters based upon said empirical measure of similarity of content of said target sets of user interest

characteristics for a virtual community of users contained in each said cluster; and

repeating said step of subdividing to produce a multi-level hierarchy of identified clusters.

10. The method of claim 9 wherein said step of generating a hierarchical menu comprises:

ascribing a cluster profile to each cluster produced by all steps of dividing and subdividing in said step of sorting.

13. Apparatus for cataloging a plurality of target objects that are stored on an electronic storage media, where users are connected via user terminals and bidirectional data communication connections to a target server that accesses said electronic storage media, said apparatus comprising:

means for storing on said electronic storage media each target; and

means for automatically generating in said target server, target profiles for each of said target objects and that are stored on said electronic storage media, each of said target profiles being generated from the contents of an associated one of said target objects their associated target object characteristics, comprising:

means for automatically generating a hierarchical menu that directs said users to at least a subset of said plurality of target objects stored on said electronic media, comprising:

means for sorting all target objects in said subset into a plurality of clusters of target objects based on an empirical measure of similarity of content of said target objects, and

means for generating a hierarchical menu that identifies the content in common of target objects sorted into each of said plurality of clusters, to enable said identified user to identify ones of said plurality of target objects stored on said electronic storage media that are likely to be of interest to said identified user.

14. The apparatus of claim 13 wherein said means for automatically generating a hierarchical menu further comprises:

means for ascribing a cluster profile to each of said plurality of clusters.

15. The apparatus of claim 13 wherein said means for sorting comprises:

means for dividing said plurality of target objects into at least two clusters based upon said empirical measure of similarity of content of said target objects;

means for subdividing each of said at least two clusters into at least two subclusters based upon said empirical measure of similarity of content of said target objects contained in each said cluster; and

means for repeating said cluster subdividing to produce a multi-level hierarchy of identified clusters.

16. The apparatus of claim 15 wherein said means for generating a hierarchical menu comprises:

means for ascribing a cluster profile to each cluster produced by all steps of dividing and subdividing in said step of sorting.

19. Apparatus for cataloging a plurality of target objects that are stored on an electronic storage media, where users are connected via user terminals and bidirectional data communication connections to a target server that accesses said electronic storage media, said apparatus comprising:

means for storing on said electronic storage media each target; and

means for automatically generating in said target server, target profiles for each of said target objects and that are stored on said electronic storage media, each of said target profiles being generated from the contents of an associated one of said target

objects their associated target object characteristics, comprising:

means for automatically generating a hierarchical menu that directs a requesting one of said users to at least a subset of said sets of user interest characteristics for a virtual community of users stored on said electronic media, comprising:

means for sorting all sets of user interest characteristics for a virtual community of users in said subset into a plurality of clusters of target sets of user interest characteristics for a virtual community of users based on an empirical measure of similarity of content of said target sets of user interest characteristics for a virtual community of users, and

means for generating a hierarchical menu that identifies the content in common of target sets of user interest characteristics for a virtual community of users sorted into each of said plurality of clusters, to enable said requesting user to identify ones of said plurality of target sets of user interest characteristics for a virtual community of users stored on said electronic storage media that are likely to be of interest to said requesting user.

20. The apparatus of claim 19 wherein said means for automatically generating a hierarchical menu further comprises:

means for ascribing a cluster profile to each of said plurality of clusters.

21. The apparatus of claim 19 wherein said means for sorting comprises:

means for dividing said plurality of target sets of user interest characteristics for a virtual community of users into at least two clusters based upon said empirical measure of similarity of content of said target sets of user interest characteristics for a virtual community of users;

means for subdividing each of said at least two clusters into at least two subclusters based upon said empirical measure of similarity of content of said target sets of user interest characteristics for a virtual community of users contained in each said cluster; and

means for repeating said step of subdividing to produce a multi-level hierarchy of identified clusters.

22. The apparatus of claim 21 wherein said means for generating a hierarchical menu comprises:

means for ascribing a cluster profile to each cluster produced by all steps of dividing and subdividing in said step of sorting.

L9: Entry 14 of 35 File: USPT Aug 20, 2002

DOCUMENT-IDENTIFIER: US 6438591 B1
TITLE: Entity management system

Detailed Description Text (18):
Another functional module 11 may, for example, function as a historical data recorder functional module 11 to periodically poll various entities in the complex system to determine their values at specific times and establish and maintain a database of the times and values to facilitate generation of usage statistics.

Detailed Description Text (29):
Alternatively, a request may solicit information as to the status or condition of one or more entities in the system, the entities being identified in the request. In processing such a request, one or more access modules 12 may determine the status or condition of the entities, and return an identification thereof to the functional-access kernel 14. In other cases, information stored in the control arrangement (such as by a historical data recorder functional module) may be used to 'satisfy the request.

Detailed Description Text (74):
The body portion 45 of the management specification contains the actual management specification for the entity. The body portion 45 is further defined in FIG. 3A. Preliminarily, the control arrangement includes two general types of entities, namely, a global entity, and a subordinate entity. The control arrangement facilitates a hierarchy of entities, as defined above, with the global entity identifying a top level entity in a hierarchy and a subordinate entity identifying a entity that is subordinate to another entity in the hierarchy. The body portion 45 of a management specification includes one of two types of entity definitions, that is, a definition 45A to a global entity or a definition 45C to a subordinate entity.

Detailed Description Text (76):
The definitions 45A and 45C to a global and subordinate entity, respectively, are further defined in FIGS. 3A through 3D. An entity definition 46 includes a name field 47 that includes a name and a code by which the entity can be identified. In addition, the name field 47 identifies the entity as a global or subordinate entity and identifies a class name for the entity. If the entity definition is for a subordinate entity, it has a superior field 50 which identifies the superior entities in the hierarchy. An identifier field 51 includes a list of attribute names for attributes which are defined later in an entity body portion 53. Finally, a symbol field 52 includes a symbol that is used to generate a specific compiler constants file which contains consistent names for use by an entity developer.

Detailed Description Text (81):
The aggregation list 55 identifies and groups all attributes having similar function. For example, an access module for a NODE4 global entity class may define an attribute aggregation called "SQUERGE". The SQUERGE attribute aggregation may include all attributes relating to the current operational performance of a NODE4 class entity, e.g., a counter type attribute indicating the number of bytes sent, and characteristic type attribute indicating the pipeline quota. In this example, a user could then view these statistics together by a command such as: SHOW NODE<instance> ALL SQUERGE

Detailed Description Text (83):
The attribute partition definition list 54 includes one or more attribute definitions 64 as further defined on FIG. 3B. Each attribute partition definition 64 includes a kind field 56 which identifies the attribute as being of a particular type, including an identifier type attribute, a status type attribute, a counter type attribute, a characteristic type attribute, a reference type attribute or a statistic type

attribute. For each type of attribute, the data type is provided by an appended field 68. The attribute partition definition 54 may also include fields 60 and 61 which indicate, respectively, a default polling rate and a maximum polling rate for the entity. As noted above, a historical data recorder functional module 11 may periodically obtain status and condition information for storage in the data storage element 17, 22 in connection with the various entities comprising the complex system. The contents of the polling rate fields identify the default and maximum rates at which the respective entities will provide status and condition information. In addition, an attribute definition includes one or more attribute fields 62 each including an attribute name 63, which includes a code by which the attribute may be accessed, and an associated attribute body 64.

Detailed Description Text (102):
When a management module is enrolled, its management specification may define new global entity classes, subentity classes or attributes, directives or events of global or subentities. The management specification (FIGS. 3A through 3D) is used to construct a data dictionary, which, in turn, is used in constructing other data structures, which are described below in connection with FIGS. 5, 8A and used as depicted in FIG. 9. The data dictionary comprises a hierarchical database having the general schema or structure shown in FIG. 4. With reference to FIG. 4, the schema has a relative root node 220 which is associated with a global entity as defined in the management specification (FIG. 3A). The global entity node points to a plurality of subsidiary nodes in the hierarchical schema, including a subsidiary node 221 listing all attributes, subsidiary node 219 listing attribute partitions, a subsidiary node 222 listing attribute aggregations, a subsidiary node 223 listing directives, and a subsidiary node 224 listing subentities, of the entity body 53 in the entity definition 46 of the management specification.

Detailed Description Text (116):
In addition to the above features, in one embodiment, the configuration database may be used in conjunction with presentation modules to support wildcarding in user commands. When a user command containing a wildcard is received by a presentation module, the presentation module issues a request to the configuration functional module, requesting an enumeration of all entities in the configuration that match the wildcard request. The configuration functional modules then uses the information in the configuration database (along with domain information) to produce the list. After receiving the list, the presentation module expands the user request into all of the possible subsidiary requests which match the wildcarding.

Detailed Description Text (135):
For example, the class data in the data dictionary (FIG. 4) indicates all of the directives 223 supported by entities in the complex system. However, the directives 223 are stored in a hierarchical format, and are subordinate to the entity classes 220. Although this format is logical for representing entity class information, it is less useful for a parse table. A user request typically lists the directive first (e.g. "SHOW" in "SHOW NODE FOO"), thus a parse table should have directives as the first level of a hierarchical structure. As can also be seen by the above example, a parse table may need to parse a command where class names (e.g. "NODE") are mixed with instance names (e.g. the identifier FOO in "NODE FOO"). Therefore, after a listing of the available directives, the parse table should list the class names which support those directives, and then the data types of instances of those classes. Although the class and data type information is available from a reorganization of the Data Dictionary, for expansion of wildcards, instance data can be obtained from the Configuration Database. Thus the parse tables in the user interface information file can consolidate directive and entity class, making the parsing of user input computationally more efficient.

Detailed Description Text (160):
Other qualifiers may be used as a distinct parameter of the request. For example, communications qualifiers include: a "TO<filename>" qualifier which sends the response of a request to a file named<filename>; a "FROM <filename>" qualifier which retrieves other request parameters from a file named<filename>; a "VIA PATH" qualifier which specifies a series of "hops" along a path, through a hierarchy of management modules (useful in specifying, e.g., the precise management module among several arrangements that will perform the operation); and a "VIA PORT" qualifier which specifies a particular network path a management module uses when performing the operation (useful, e.g., to specify that an access module will perform a diagnostic test using a specific EtherNet port.)

Detailed Description Text (164):
where the time argument<time-arg>may, e.g., indicate the start time ("START=<time>"), the end time ("END=<time>") or duration ("DURATION=<time-length>"), the period of repetition ("REPEAT EVERY[=]<time-length>"), the time accuracy ("CONFIDENCE[=]<time-length>), or the sampling rate ("SAMPLE RATE[=]<time-length>"). These arguments may interact with one another to create a general schedule and scope of interest for a request. In particular, in one particular embodiment, the three time arguments, START, END and DURATION are related such that any two of them define a period. Thus when a time-normalized entity statistic is displayed, at least two of these qualifier arguments must be specified.

Detailed Description Text (172):
Scope of interest time specifications are supplied by requests using the time specifier field 123. Using a time specifier, other values of data than "the value it has right now" can be displayed and processed, and statistics can be computed over some time period. In one particular embodiment, a time "scope of interest" is expressed by prepositional phrases in the time specifier of a request. Generally, a time specifier is used with a SHOW command, but time contexts may also apply to MODIFY type requests and actions.

Detailed Description Text (184):
As discussed above, if a request can be satisfied by a single response, the response is generated and returned to the requestor. In the more general case, the service provider, e.g., a functional module, information manager, or an access module, cannot satisfy the request in one reply. For example, the requester may have used wildcarding in the input entity parameter 121, to specify a group of entities. As each reply can only incorporate information from a single entity, several replies are required, one for each entity. In another case, a request to a single entity may have a time specifier with several different time values. As each reply can only incorporate information for a single time value, several replies are required, one for each time. A request that requires multiple replies can be for any type of operation, including obtaining attribute data about an entity or entities, modifying attributes of several entities, and modifying the state of several entities.

Detailed Description Text (200):
The term "entity node" is used to describe the data structure 130 because it satisfies the entity model set forth above. Generally, data structure 130 satisfies the entity model because it has a hierarchical structure and its child structures resemble it. The term "entity node" as it is used to describe data structure 130 should not be confused with the term "entity" used to describe elements of the complex system.

Detailed Description Text (246):
In alternative embodiments, a first domain may incorporate the members of a second domain by reference to the second domain, thus reducing the size of the domains database. In other embodiments, the domains database may establish a hierarchy of domains similar to the hierarchy of entities and subentities, and commands may be directed similarly to domains and subdomains.

Detailed Description Text (247):
The configuration database includes an entry 234 for each entity and subentity, organized hierarchically in the database. The full name for each entity and subentity instance is provided. This information can be used by the configuration functional module to quickly determine the configuration, for example, to display (via a presentation module) to the user a map of the configuration or menus of entity instance names.

WEST

Generate Collection Print

L9: Entry 16 of 35 File: USPT Feb 22, 2000

DOCUMENT-IDENTIFIER: US 6029195 A
TITLE: System for customized electronic identification of desirable objects

Brief Summary Text (15):
Relevant definitions of terms for the purpose of this description include: (a.) an object available for access by the user, which may be either physical or electronic in nature, is termed a "target object", (b.) a digitally represented profile indicating that target object's attributes is termed a "target profile", (c.) the user looking for the target object is termed a "user", (d.) a profile holding that user's attributes, including age/zip code/etc. is termed a "user profile", (e.) a summary of digital profiles of target objects that a user likes and/or dislikes, is termed the "target profile interest summary" of that user, (f.) a profile consisting of a collection of attributes, such that a user likes target objects whose profiles are similar to this collection of attributes, is termed a "search profile" or in some contexts a "query" or "query profile," (g.) a specific embodiment of the target profile interest summary which comprises a set of search profiles is termed the "search profile set" of a user, (h.) a collection of target objects with similar profiles, is termed a "cluster," (i.) an aggregate profile formed by averaging the attributes of all tar get objects in a cluster, termed a "cluster profile," (j.) a real number determined by calculating the statistical variance of the profiles of all target objects in a cluster, is termed a "cluster variance," (k.) a real number determined by calculating the maximum distance between the profiles of any two target objects in a cluster, is termed a "cluster diameter."

Brief Summary Text (21):
The ability to measure the similarity of profiles describing target objects and a user's interests can be applied in two basic ways: filtering and browsing. Filtering is useful when large numbers of target objects are described in the electronic medias pace. These target objects can for example be articles that are received or potentially received by a user, who only has time to read a small fraction of them. For example, one might potentially receive all items on the AP news wire service, all items posted to a number of news groups, all advertisements in a set of newspapers, or all unsolicited electronic mail, but few people have the time or inclination to read so many articles. A filtering system in the system for customized electronic identification of desirable objects automatically selects a set of articles that the user is likely to wish to read. The accuracy of this filtering system improves over time by noting which articles the user reads and by generating a measurement of the depth to which the user reads each article. This information is then used to update the user's target profile interest summary. Browsing provides an alternate method of selecting a small subset of a large number of target objects, such as articles. Articles are organized so that users can actively navigate among groups of articles by moving from one group to a larger, more general group, to a smaller, more specific group, or to a closely related group. Each individual article forms a one-member group of its own, so that the user can navigate to and from individual article s as well as larger groups. The methods used by the system for customized electronic identification of desirable objects allow articles to be grouped into clusters and the clusters to be grouped and merged into larger and larger clusters. These hierarchies of clusters then form the basis for menuing and navigational systems to allow the rapid searching of large numbers of articles. This same clustering technique is applicable to any type of target objects that can be profiled on the electronic media such as product selections within a menu or throughout the World Wide Web.

Drawing Description Text (5):
FIG. 5 illustrates in flow diagram form a method for automatically generating article profiles and an associated hierarchical menu system;

Drawing Description Text (8):
FIG. 11 illustrates a hierarchical cluster tree example;

Detailed Description Text (118):
Hierarchical clustering of target objects is often useful. Hierarchical clustering produces a tree which divides the target objects first into two large clusters of roughly similar objects; each of these clusters is in turn divided into two or more smaller clusters, which in turn are each divided into yet smaller clusters until the collection of target objects has been entirely divided into "clusters" consisting of a single object each, as diagrammed in FIG. 8 In this diagram, the node d denotes a particular target object d, or equivalently, a single-member cluster consisting of this target object. Target object d is a member of the cluster (a, b, d), which is a subset of the cluster (a, b, c, d, e, f), which in turn is a subset of all target objects. The tree shown in FIG. 8 would be produced from a set of target objects such as those shown geometrically in FIG. 7. In FIG. 7, each letter represents a target object, and axes x1 and x2 represent two of the many numeric attributes on which the target objects differ. Such a cluster tree may be created by hand, using human judgment to form clusters and subclusters of similar objects, or may be created automatically in either of two standard ways: top-down or bottom-up. In top-down hierarchical clustering, the set of all target objects in FIG. 7 would be divided into the clusters (a, b, c, d, e, f) and (g, h, i, j, k). The clustering algorithm would then be reapplied to the target objects in each cluster, so that the cluster (g, h, i, j, k) is subpartitioned into the clusters (g, k) and (h, i, j), and so on to arrive at the tree shown in FIG. 8. In bottom-up hierarchical clustering, the set of all target objects in FIG. 7 would be grouped into numerous small clusters, namely (a, b), d, (c, f), e, (g,k), (h, i), and j. These clusters would then themselves be grouped into the larger clusters (a, b, d), (c, e, f), (g, k), and (h, i, j), according to their cluster profiles. These larger clusters would themselves be grouped into (a, b, c, d, e, f) and (g, k, h, i, j), and so on until all target objects had been grouped together, resulting in the tree of FIG. 8. Note that for bottom-up clustering to work, it must be possible to apply the clustering algorithm to a set of existing clusters. This requires a notion of the distance between two clusters. The method disclosed above for measuring the distance between target objects can be applied directly, provided that clusters are profiled in the same way as target objects. It is only necessary to adopt the convention that a cluster's profile is the average of the target profiles of all the target objects in the cluster; that is, to determine the cluster's value for a given attribute, take the mean value of that attribute across all the target objects in the cluster. For the mean value to be well-defined, all attributes must be numeric, so it is necessary as usual to replace each textual or associative attribute with its decomposition into numeric attributes (scores), as described earlier. For example, the target profile of a single Woody Allen film would assign "Woody-Allen" a score of 1 in the "name-of-director" field, while giving "Federico-Fellini" and "Terence-Davies" scores of 0. A cluster that consisted of 20 films directed by Allen and 5 directed by Fellim would be profiled with scores of 0.8, 0.2, and 0 respectively, because, for example, 0.8 is the average of 20 ones and 5 zeros.

Detailed Description Text (120):
Given a target object with target profile P, or alternatively given a search profile P, a hierarchical cluster tree of target objects makes it possible for the system to search efficiently for target objects with target profiles similar to P. It is only necessarily to navigate through the tree, automatically, in search of such target profiles. The system for customized electronic identification of desirable objects begins by considering the largest, top-level clusters, and selects the cluster whose profile is most similar to target profile P. In the event of a near-tie, multiple clusters may be selected. Next, the system considers all subclusters of the selected clusters, and this time selects the subclusters or subclusters whose profiles are closest to target profile P. This refinement process is iterated until the clusters selected on a given step are sufficiently small, and these are the desired clusters of target objects with profiles most similar to target profile P. Any hierarchical cluster tree therefore serves as a decision tree for identifying target objects. In pseudo-code form, this process is as follows (and in flow diagram form in FIGS. 13A and 13B):

Detailed Description Text (122):
2. Initialize the current tree T to be the hierarchical cluster tree of all objects at step 13A01 and at step 13A02 scan the current cluster tree for target objects similar to P, using the process detailed in FIG. 13B. At step 13A03, the list of target objects is returned.

Detailed Description Text (128):

In step 5 of this pseudo-code, smaller thresholds are typically used at lower levels of the tree, for example by making the threshold an affine function or other function of the cluster variance or cluster diameter of the cluster p.sub.i. If the cluster tree is distributed across a plurality of servers, as described in the section of this description titled "Network Context of the Browsing System", this process may be executed in distributed fashion as follows: steps 3-7 are executed by the server that stores the root node of hierarchical cluster tree T, and the recursion in step 7 to a subcluster tree T.sub.i involves the transmission of a search request to the server that stores the root node of tree T.sub.i, which server carries out the recursive step upon receipt of this request. Steps 1-2 are carried out by the processor that initiates the search, and the server that executes step 6 must send a message identifying the target object to this initiating processor, which adds it to the list.

Detailed Description Text (129):
Assuming that low-level clusters have been already been formed through clustering, there are alternative search methods for identifying the low-level cluster whose profile is most similar to a given target profile P. A standard back-propagation neural net is one such method: it should be trained to take the attributes of a target object as input, and produce as output a unique pattern that can be used to identify the appropriate low-level cluster. For maximum accuracy, low-level clusters that are similar to each other (close together in the cluster tree) should be given similar identifying patterns. Another approach is a standard decision tree that considers the attributes of target profile P one at a time until it can identify the appropriate cluster. If profiles are large, this may be more rapid than considering all attributes. A hybrid approach to searching uses distance measurements as described above to navigate through the top few levels of the hierarchical cluster tree, until it reaches an cluster of intermediate size whose profile is similar to target profile P, and then continues by using a decision tree specialized to search for low-level subclusters of that intermediate cluster.

Detailed Description Text (130):
One use of these searching techniques is to search for target objects that match a search profile from a user's search profile set. This form of searching is used repeatedly in the news clipping service, active navigation, and Virtual Community Service applications, described below. Another use is to add a new target object quickly to the cluster tree. An existing cluster that is similar to the new target object can be located rapidly, and the new target object can be added to this cluster. If the object is beyond a certain threshold distance from the cluster center, then it is advisable to start a new cluster. Several variants of this incremental clustering scheme can be used, and can be built using variants of subroutines available in advanced statistical packages. Note that various methods can be used to locate t he new target objects that must be added to the cluster tree, depending on the architecture used. In one method, a "webcrawler" program running on a central computer periodically scans all servers in search of new target objects, calculates the target profiles of these objects, and adds them to the hierarchical cluster tree by the above method. In another, whenever a new target object is added to any of the servers, a software "agent" at that server calculates the target profile and adds it to the hierarchical cluster tree by the above method.

Detailed Description Text (137):
Similar information can alternatively be extracted from a collection of consumer profiles without recourse to a decision tree, by considering attributes one at a time, and identifying those attributes on which product X's consumers differ significantly from its non-consumers. These techniques serve to characterize consumers of a particular product; they can be equally well applied to voter research or other survey research, where the objective is to characterize those individuals from a given set of surveyed individuals who favor a particular candidate, hold a particular opinion, belong to a articular demographic group, or have some other set of distinguishing attributes. Researchers may wish to purchase batches of analyzed or unanalyzed user profiles from which personal identifying information has been removed. As with any statistical database, statistical conclusions can be drawn, and relationships between attributes can be elucidated using knowledge discovery techniques which are well known in the art.

Detailed Description Text (139):
In the case of profiling new products, a decision tree may be useful for determining its profile quickly (for example if certain general attributes are known about the product). Rapid profiling may also be used to automatically present a selection of attributes (of at least two) with which a user selects which attribute most aptly

describes the product and/or provides a weighted value of its relevance thereto. Alternatively, the decision tree presents (for each node) at least one exemplar item which the user rates indicating the degree of similarity between the system presented item(s) and the new item of interest. Additionally, for the sake of optimizing the confidence of the users being surveyed, the decision tree may also identify the user whose profiles suggest the greatest degree of similarity with the attributes or items being presented as queries. In one variation in this regard, the system selects users which are most familiar with two or more competitive products. The system performs a rapid profiling of these users, however, for product attributes which are most relevant to both products (which is produced from the result of combining or averaging both product profiles). Example attributes which are most telling about the user's perception of comparative value and quality when making a selection may include: performance, aesthetics, comfort, convenience of use, value, overall satisfaction, personal preference, as well as other relevant specific product attributes which may be determined as a part of the user's profile. By applying this technique over multiple product brands within a given category, a relative, comparative measure can be determined through averaging of results across all participating users on an attribute specific basis. Using the techniques described above which allow for pseudonymous credentialing of users or organizations by other entities, these evaluation based attributes may be automatically ascribed to each product in the form of credentials, also manually ascribed comments or descriptions may be (provided and subsequently rated by other users) to further leverage consumer participation in adding characterization attributes to a given product's or entities profile. These averaged consumer rating based credentials also act as a means of normalizing biased opinions or rogue attempts to defame a product or entity and thus are used to substantiate claims which consumers have provided and other consumers have substantiated either in the form of on-line or off-line advertisements and coupons. Comparative ratings of competitive products are achievable by targeting users which have experience with (two or more) products being compared. The most relevant attributes which both products share are presented using these rapid profiling techniques. In order to develop a truly robust statistically confident comparison across all products on an attribute by attribute basis, it is important to use this comparative product rating approach, to identify automatically which product comparisons are most statistically relevant in order to provide statistical confidence for all products being evaluated (in this comparative product context) to validation of the values of each attribute using different combinations of product comparisons is important in order to assure statistical confidence (between different users). These rated attribute credentials may also be segmented by user types using knowledge discovery techniques. For example, it is possible that users of a certain demographic, product affinity or other attribute type may have different preferences demands or expectations, thus may evaluate a product's overall quality or value (or other product attribute) differently. Additionally, these credentials may be provided as resolution credentials, for example in combination with a credential provided by a neutral third party which proves that the user is in good standing with its customers (that a "significant" number of complaints were not submitted). Brokerage exchanges which match buyers and sellers and/or act as a directory thereof may wish to apply these techniques in order to provide users with some unbiased feedback from peers about products and services being solicited peer to peer rating based resolution credentials. It is also possible to automatically present a set of survey questions to a group of users who have been previously interacting on-line with another user. Because of the subjective nature involved in characterizing individuals based upon their personal, or even professional proficiencies and weaknesses, human involvement in providing manual characterizations of a sample of users is necessary. The nature of the interaction (an associate, professional, personal, or social) may be determined through automatic means (based on the content profiles of dialogues and lists of "similar" users which they interact with) in order to automatically ascribe an associative attribute which identifies both other individuals, his/her relationship with the user and the nature of their interaction. Individuals may be automatically presented with targeted questions appropriate to the nature thereof in accordance with their mutual relationship through anticipation of which attributes or queries other individuals (like friends, associates, business partners or employers) are most likely to request in the future. These questions are ideally requested from multiple users, their values are then averaged and may be ascribed to that user as resolution credentials. In case of disputes mediation by a judicating third party may be required. Additionally, the system may further anticipate the types of questions which are most likely to be requested by other users in the future. This approach may also be used by the system to profile skills sets, qualifications, issues of personality, character or qualification to perform a particular task. It may also direct queries to the users most likely to be qualified knowledgeable in certain popular domains, which are most likely to be relevant (and thus anticipate the types of queries that other users are likely to

request. Similarly, users may be used to answer questions or provide descriptive characterizations of certain tasks or queries using rapid profiling in this way as well. Thus, tasks, (consulting on the internet, intranet, etc.) may be profiled according to the types of users who ascribe, subjective, or objective attributes to best describe the task, or attributes may be ascribed which characterize the most appropriate individuals according to their professional qualifications or other relevant attributes, such as the tasks which they have successfully performed. Accordingly, task attributes may also be conveyed to the best candidates to whom these tasks are directed. As suggested, task performance may be manually evaluated in order to provide the system with a source of performance based relevance feedback. The users who submitted the task offers are given the opportunity to provide an evaluation of the level of the quality of the work (or query response) as well as overall satisfaction regarding the response to the request offer. The requester may provide an evaluation in the form of a set of feedback comments. Additionally, the rapid profiling technique will automatically generate a set of the most relevant attributes in the form of a survey which allow the user to rate the attributes according to each relevant attribute parameter as perceived by the user. (These attributes may, of course, include those which are humanly ascribed as well). Unlike the method for automatic query routing the current system for finding optimal user skill profiles to match the particular submitted task description, the current system potentially embodies a much more complex knowledge construction requiring precision-oriented statistical knowledge about the nature of the user's numerous skill sets and the submitted tasks.

Detailed Description Text (140):
It may be very useful to use associative attributes to identify the relevant words in the task description and users who successfully provided solutions and responses to similarly described tasks in the past. According to the previously described techniques of the patent, the collection of target objects in this particular information domain include task descriptions; solutions to the requests, individuals who have provided solutions to those tasks, individuals whose profiles qualify them for solving particular problem types, and individuals who are most likely to have a need for solution to a particular type of problem. As suggested each of these types of target objects may constitute the information space of the presently described system for customized electronic identification of desirable objects. Thus in order to augment the search retrieval process the user may also be directed to potentially useful information through, menu browsing and search query navigation (and nearest neighbor, target object to target object) navigation down or across the menu as well as the current matching of appropriate users with requests are herein described. Accordingly, as relevant in the other informational domains (if the target object profiles) and the similarity between target objects is not statistically confident the system will cross correlate the statistical data from other informational domains in order to assign the most appropriate profile for each of target object for which a sparse data problem currently exists. In a more advanced embodiment, profiling of target objects in this complex domain may be further enhanced by establishing exception in the form of special appropriateness function rules between the textual, descriptive, and numeric attributes of those targeted objects (e.g. the qualification of the users, the textual attributes in the description of each task, and the evaluative description of the recipients of the task solutions provided. As in other informational domains, the exception rules which apply to a particular domain are given priority over those which apply to another domain. (Again, where cross correlation statistics are given second priority in order to maximize statistical confidence). Such exception rules may include (but are not limited to) giving special relevance between a word attribute based upon the sequence in which those textual attributes appear in the description, (or in the presence or absence of a numeric attribute in combination with a numeric attribute or a textual attribute). (These associations may also be based on their relative frequencies in the text as well) or more complex rules may be established automatically. Furthermore, if the combination of words appear, and the request is from a particular user it is likely that a particular detailed target profile is appropriate for the target object. By definition, exception rules apply exceptions in the weighting values of attributes or an attribute with an exception is present (or at least one of) at least three attributes which are present in a particular (user or target object) profile whose attribute weighting influence upon another attribute would not otherwise be recognized in a pure (non-rule based) statistical model (customized) profiles of requests which is specific to each user may be used as each user may submit similar requests in a different descriptive manner (with varying word usage). The user's needs may also vary based upon the context of what actions the user has recently performed e.g., searching through particular topics of the World Wide Web, searching through e-mail, conversing with particular users about a particular topic of engaging in these activities at certain times or in conjunction with any of the above which may indicate the context of

the user's mode of activities such as work, leisure or academics. If a particular combination of words appears and it is from a particular request as part of the description of a request from a particular individual, the relevance of each attribute component of the request may be different to some degree than the request from a different individual (wherein this case these exception rules are relevant to particular users). Accordingly, the sequence of words which appear (for a particular word combination) may be suggestive of the relative importance of particular words to one another or to a particular solution or a particular individual. Accordingly in the application to matching queries or tasks with users according to their qualifications for the particular combination of qualifying credentials which a user possesses may indicate an exception rule either between particular credentials, between credentials and individual tasks (or between credentials and textual attributes in the text of task descriptions). Exception rules are not applicable for associative attributes which associate target objects users (or both) via the present similarity based techniques.

Detailed Description Text (149):
In accordance with the techniques presently suggested, just as categories of information contain profiles, the most appropriate information (e.g., news information) can be automatically routed to the most appropriate category. Similarly content may be automatically routed to the most appropriate virtual channels which appeal to a particular type of audience (not only based on its content, but more subjective criteria as well) offering a unique multi media experience, writing or commentary style of its authors, etc. For this reason it may be most appropriate to initially gather relevance feedback of which users access the information in order to develop statistical confidence as to its associative attributes before it is routed to a particular channel. For example, in this regard as with the presently described techniques for customizing content through indexing, navigation and delivery from the entire scope of available information on the Internet, the scope of information may be narrowed to that of a particular channel. Additionally, because considerable overlap of content may occur between channels, authors and editors of a particular channel may use this technique to select the most desirable content from which appropriate editing and revisions may be performed as desired. These channels ideally are presented in combination with virtual communities (e.g., virtual text and voice chat rooms). They may accordingly be navigated to/from as part of the 3-D representation of the surrounding information space. For example virtual chat room associated with a news channel may incorporate scheduled live interviews with news reporters (or news makers) who had covered (or had been involved in) a particular story or combination of stories during which time participants may submit questions or comments (pseudonymously if desired). Polls may be taken about these users views on each particular event or controversial issues that are newsworthy. As suggested, preference based attributes, demographics and psychological user attributes may be statistically correlated with certain news from survey question responses or as otherwise submitted (such as in the form of active comments about that particular issue). Because questions and comments from many users may bombard a particular chat room, automated methods may be used to more efficiently manage large quantities of data. Specifically, the system may apply the following techniques:

Detailed Description Text (150):
1. Real time automatic identification of similar queries or comments which had been previously submitted (using statistical NLP or deeper NLU techniques). Once a user has submitted a question or comment, the system instantaneously indexes any similar item(s) previously submitted, automatically notifies the user that the user's submission has been canceled and automatically retrieves the previously submitted response to that previously submitted item. In the context of an ascribed posting to news groups currently known techniques such as auto-FAQ are able to generate FAQs automatically. For either live chat or (asynchronous) newsgroups, this technique may instead be used to eliminate redundancy by identifying (by indexing in real time via statistical NLP) pre-existing similar correspondences to those which are about to be initiated.

Detailed Description Text (178):
In other scenarios, the request R to proxy server S2 formed by the user may have different content. For example, request R may instruct proxy server S2 to use the methods described later in this description to retrieve from the most convenient server a particular piece of information that has been multicast to many servers, and to send this information to the user. Conversely, request R may instruct proxy server S2 to multicast to many servers a file associated with a new target object provided by the user, as described below. If the user is a subscriber to the news clipping service described below, request R may instruct proxy server S2 to forward to the user all target objects that the news clipping service has sent to proxy server S2 for the

user's attention. If the user is employing the active navigation service described below, request R may instruct proxy server S2 to select a particular cluster from the hierarchical cluster tree and provide a menu of its subclusters to the user, or to activate a query that temporarily affects proxy server S2's record of the user's target profile interest summary. If the user is a member of a virtual community as described below, request R may instruct proxy server S2 to forward to the user all messages that have been sent to the virtual community.

Detailed Description Text (193):
Pre-fetching of locally stored data has been heavily studied in memory hierarchies, including CPU caches and secondary storage (disks), for several decades. A leader in this area has been A. J. Smith of Berkeley, who identified a variety of schemes and analyzed opportunities using extensive traces in both databases and CPU caches. His conclusion was that general schemes only really paid off where there was some reasonable chance that sequential access was occurring, e.g., in a sequential read of data. As the balances between various latencies in the memory hierarchy shifted during the late 1980's and early 1990's, J. M. Smith and others identified further opportunities for pre-fetching of both locally stored data and network data. In particular, deeper analysis of patterns in work by Blaha showed the possibility of using expert systems for deep pattern analysis that could be used for pre-fetching. Work by J. M. Smith proposed the use of reference history trees to anticipate references in storage hierarchies where there was some historical data. Recent work by Touch and the Berkeley work addressed the case of data on the World-Wide Web, where the large size of images and the long latencies provide extra incentive to pre-fetch; Touch's technique is to pre-send when large bandwidths permit some speculation using HTML storage references embedded in WEB pages, and the Berkeley work uses techniques similar to J. M. Smith's reference histories specialized to the semantics of HTML data.

Detailed Description Text (194):
Successful pre-fetching depends on the ability of the system to predict the next action or actions of the user. In the context of the system for customized electronic identification of desirable objects, it is possible to cluster users into groups according to the similarity of their user profiles. Any of the well-known pre-fetching methods that collect and utilize aggregate statistics on past user behavior, in order to predict future user behavior, may then be implemented in so as to collect and utilize a separate set of statistics for each cluster of users. In this way, the system generalizes its access pattern statistics from each user to similar users, without generalizing among users who have substantially different interests. The system may further collect and utilize a similar set of statistics that describes the aggregate behavior of all users; in cases where the system cannot confidently make a prediction as to what a particular user will do, because the relevant statistics concerning that user's user cluster are derived from only a small amount of data, the system may instead make its predictions based on the aggregate statistics for all users, which are derived from a larger amount of data. For the sake of concreteness, we now describe a particular instantiation of a pre-fetching system, that both employs these insights and that makes its pre-fetching decisions through accurate measurement of the expected cost and benefit of each potential pre-fetch.

Detailed Description Text (236):
(c.) satisfying the request would involve disclosure to the accessor of a certain fact about the user's user profile

Detailed Description Text (237):
(d.) satisfying the request would involve disclosure to the accessor of the user's target profile interest summary

Detailed Description Text (238):
(e.) satisfying the request would involve disclosure to the accessor of statistical summary data, which data are computed from the useres user profile or target profile interest summary together with the user profiles and target profile interest summaries of at least n other users in the user base of the proxy server

Detailed Description Text (265):
A set of topical multicast trees for a set of homogenous target objects may be constructed or reconstructed at any time, as follows. The set of target objects is grouped into a fixed number of topical clusters C1 . . . Cp with the methods described above, for example, by choosing C1 . . . Cp to be the result of a k-means clustering of the set of target objects, or alternatively a covering set of low-level clusters from a

hierarchical cluster tree of these target objects. A multicast tree MT(c) is then constructed from each cluster C in C1 . . . Cp, by the following procedure:

Detailed Description Text (300):
These news articles are then hierarchically clustered in a hierarchical cluster tree at step 503, which serves as a decision tree for determining which news articles are closest to the user's interest. The resulting clusters can be viewed as a tree in which the top of the tree includes all target objects and branches further down the tree represent divisions of the set of target objects into successively smaller subclusters of target objects. Each cluster has a cluster profile, so that at each node of the tree, the average target profile (centroid) of all target objects stored in the subtree rooted at that node is stored. This average of target profiles is computed over the representation of target profiles as vectors of numeric attributes, as described above.

Detailed Description Text (302):
The process by which a user employs this apparatus to retrieve news articles of interest is illustrated in flow diagram form in FIG. 11. At step 1101, the user logs into the data communication network N via their client processor C.sub.1 and activates the news reading program. This is accomplished by the user establishing a pseudonymous data communications connection as described above to a proxy server S.sub.2, which provides front-end access to the data communication network N. The proxy server S.sub.2 maintains a list of authorized pseudonyms and their corresponding public keys and provides access and billing control. The user has a search profile set stored in the local data storage medium on the proxy server S.sub.2. When the user requests access to "news" at step 1102, the profile matching module 203 resident on proxy server S.sub.2 sequentially considers each search profile p.sub.k from the user's search profile set to determine which news articles are most likely of interest to the user. The news articles were automatically clustered into a hierarchical cluster tree at an earlier step so that the determination can be made rapidly for each user. The hierarchical cluster tree serves as a decision tree for determining which articles' target profiles are most similar to search profile p.sub.k : the search for relevant articles begins at the top of the tree, and at each level of the tree the branch or branches are selected which have cluster profiles closest to p.sub.k. This process is recursively executed until the leaves of the tree are reached, identifying individual articles of interest to the user, as described in the section "Searching for Target Objects" above.

Detailed Description Text (303):
A variation on this process exploits the fact that many users have similar interests. Rather than carry out steps 5-9 of the above process separately for each search profile of each user, it is possible to achieve added efficiency by carrying out these steps only once for each group of similar search profiles, thereby satisfying many users' needs at once. In this variation, the system begins by non-hierarchically clustering all the search profiles in the search profile sets of a large number of users. For each cluster k of search profiles, with cluster profile p.sub.k, it uses the method described in the section "Searching for Target Objects" to locate articles with target profiles similar to p.sub.k. Each located article is then identified as of interest to each user who has a search profile represented in cluster k of search profiles.

Detailed Description Text (304):
Notice that the above variation attempts to match clusters of search profiles with similar clusters of articles. Since this is a symmetrical problem, it may instead be given a symmetrical solution, as the following more general variation shows. At some point before the matching process commences, all the news articles to be considered are clustered into a hierarchical tree, termed the "target profile cluster tree," and the search profiles of all users to be considered are clustered into a second hierarchical tree, termed the "search profile cluster tree." The following steps serve to find all matches between individual target profiles from any target profile cluster tree and individual search profiles from any search profile cluster tree: 1. For each child subtree S of the root of the search profile cluster tree (or, let S be the entire search profile cluster tree if it contains only one search profile): 2. Compute the cluster profile P.sub.S to be the average of all search profiles in subtree S 3. For each subcluster (child subtree) T of the root of the target profile cluster tree (or, let T be the entire target profile cluster tree if it contains only one target profile): 4. Compute the cluster profile P.sub.T to be the average of all target profiles in subtree T 5. Calculate d(P.sub.S, P.sub.T, the distance between P.sub.S and P.sub.T 6. If d(P.sub.S, P.sub.T)<t, a threshold, 7. If S contains only one search profile and T contains only one target profile, declare a match between that search profile and that target profile, 8. otherwise recurse to step 1 to find all matches

between search profiles in tree S and target profiles in tree T.

Detailed Description Text (332):
A hierarchical cluster tree imposes a useful organization on a collection of target objects. The tree is of direct use to a user who wishes to browse through all the target objects in the tree. Such a user may be exploring the collection with or without a well-specified goal. The tree's division of target objects into coherent clusters provides an efficient method whereby the user can locate a target object of interest. The user first chooses one of the highest level (largest) clusters from a menu, and is presented with a menu listing the subclusters of said cluster, whereupon the user may select one of these subclusters. The system locates the subclusters, via the appropriate pointer that was stored with the larger cluster, and allows the user to select one of its subclusters from another menu. This process is repeated until the user comes to a leaf of the tree, which yields the details of an actual target object. Hierarchical trees allow rapid selection of one target object from a large set. In ten menu selections from menus of ten items (subclusters) each, one can reach 10.sup.10 =10,000,000,000 (ten billion) items. In the preferred embodiment, the user views the menus on a computer screen or terminal screen and selects from them with a keyboard or mouse. However, the user may also make selections over the telephone, with a voice synthesizer reading the menus and the user selecting subclusters via the telephone's touch-tone keypad. In another variation, the user simultaneously maintains two connections to the server, a telephone voice connection and a fax connection; the server sends successive menus to the user by fax, while the user selects choices via the telephone's touch-tone keypad.

Detailed Description Text (333):
Just as user profiles commonly include an associative attribute indicating the user's degree of interest in each target object, it is useful to augment user profiles with an additional associative attribute indicating the user's degree of interest in each cluster in the hierarchical cluster tree. This degree of interest may be estimated numerically as the number of subclusters or target objects the user has selected from menus associated with the given cluster or its subclusters, expressed as a proportion of the total number of subclusters or target objects the user has selected. This associative attribute is particularly valuable if the hierarchical tree was built using "soft" or "fuzzy" clustering, which allows a subclusters or target object to appear in multiple clusters: if a target document appears in both the "sports" and the "humor" clusters, and the user selects it from a menu associated with the "humor" cluster, then the system increases its association between the user and the "humor" cluster but not its association between the user and the "sports" cluster.

Detailed Description Text (339):
It should be appreciated that a hierarchical cluster-tree may be configured with multiple cluster selections branching from each node or the same labeled clusters presented in the form of single branches for multiple nodes ordered in a hierarchy. In one variation, the user is able to perform lateral navigation between neighboring clusters as well, by requesting that the system search for a cluster whose cluster profile resembles the cluster profile of the currently selected cluster. If this type of navigation is performed at the level of individual objects (leaf ends), then automatic hyperlinks may be then created as navigation occurs. This is one way that nearest neighbor clustering navigation may be performed. For example, in a domain where target objects are home pages on the World Wide Web, a collection of such pages could be laterally linked to create a "virtual mall". Most importantly, links to sites in the form of targeted advertisements may be temporarily generated (as a result of the user profile and the target object profile of the page being visited, the dialogue being conducted or the content being viewed, listened to or read at that moment). This is one way in which "on the fly" automatic creation of customized links may occur (user specific linking of advertisers with sites or other content including programming or joint ads or promotions between advertisers may occur in real time). Or in another period this technique may be used to recommend the most befitting sites and/or ads which should be linked together (based upon their similarity). Of course, certain promotions for example may be directly competitive such as a product for two brands of toothpaste. Such direct competitive overlap must thus be accounted for. This technique may also account for one way or two way (exchanged) links between vendors. Advertisers which exchange links or wish to link to a "prime location" should pay a price which is directly in accordance with the market demand for that advertisement though not exceeding the price value necessary to fill the available ad space. The techniques described in co-pending patent application entitled "PPS" suggests a method of automatically generating a customized motion (or joint promotion) for individual users. A similar technique may be used to automatically establish a price for the ad space

(based on a combined predicted price per impression and predicted value for the average customer expected to access that advertisement. As feedback occurs, this pricing model is adjusted according to actual response feedback, links may be broken, reformed in a one way or two way context in automatic fashion as such.

Detailed Description Text (345):
Although the topology of a hierarchical cluster tree is fixed by the techniques that build the tree, the hierarchical menu presented to the user for the user's navigation need not be exactly isomorphic to the cluster tree. The menu is typically a somewhat modified version of the cluster tree, reorganized manually or automatically so that the clusters most interesting to a user are easily accessible by the user. In order to automatically reorganize the menu in a user-specific way, the system first attempts automatically to identify existing clusters that are of interest to the user. The system may identify a cluster as interesting because the user often accesses target objects in that cluster--or, in a more sophisticated variation, because the user is predicted to have high interest in the cluster's profile, using the methods disclosed herein for estimating interest from relevance feedback.

Detailed Description Text (346):
Several techniques can then be used to make interesting clusters more easily accessible. The system can at the user's request or at all times display a special list of the most interesting clusters, or the most interesting subclusters of the current cluster, so that the user can select one of these clusters based on its label and jump directly to it. In general, when the system constructs a list of interesting clusters in this way, the I.sup.th most prominent choice on the list, which choice is denoted Top(I), is found by considering all appropriate clusters C that are fairther than a threshold distance t from all of Top(1), Top(2), . . . Top(I-1), and selecting the one in which the user's interest is estimated to be highest. Here the threshold distance t is optionally dependent on the computed cluster variance or cluster diameter of the profiles in the latter cluster. Several techniques that reorganize the hierarchical menu tree are also usefil. First, menus can be reorganized so that the most interesting subcluster choices appear earliest on the menu, or are visually marked as interesting; for example, their labels are displayed in a special color or type face, or are displayed together with a number or graphical image indicating the likely level of interest. Second, interesting clusters can be moved to menus higher in the tree, i.e., closer to the root of the tree, so that they are easier to access if the user starts browsing at the root of the tree. Third, uninteresting clusters can be moved to menus lower in the tree, to make room for interesting clusters that are being moved higher. Fourth, clusters with an especially low interest score (representing active dislike) can simply be suppressed from the menus; thus, a user with children may assign an extremely negative weight to the "vulgarity" attribute in the determination of q, so that vulgar clusters and documents will not be available at all. As the interesting clusters and the documents in them migrate toward the top of the tree, a customized tree develops that can be more efficiently navigated by the particular user. If menus are chosen so that each menu item is chosen with approximately equal probability, then the expected number of choices the user has to make is minimized. If, for example, a user frequently accessed target objects whose profiles resembled the cluster profile of cluster (a, b, d) in FIG. 8 then the menu in FIG. 9 could be modified to show the structure illustrated in FIG. 10.

Detailed Description Text (348):
In a system where queries are used, it is useful to include in the target profiles an associative attribute that records the associations between a target object and whatever terms are employed in queries used to find that target object. The association score of target object X with a particular query term T is defined to be the mean relevance feedback on target object X, averaged over just those accesses of target object X that were made while a query containing term T was active, multiplied by the negated logarithm of term T's global frequency in all queries. The effect of this associative attribute is to increase the measured similarity of two documents if they are good responses to queries that contain the same terms. A further maneuver can be used to improve the accuracy of responses to a query: in the summation used to determine the quality q(U, X) of a target object X, a term is included that is proportional to the sum of association scores between target object X and each term in the active query, if any, so that target objects that are closely associated with terms in an active query are determined to have higher quality and therefore higher interest for the user. To complement the system's automatic reorganization of the hierarchical cluster tree, the user can be given the ability to reorganize the tree manually, as he or she sees fit. Any changes are optionally saved on the user's local storage device so that they will affect the presentation of the tree in future sessions. For example, the

user can choose to move or copy menu options to other menus, so that useful clusters can thereafter be chosen directly from the root menu of the tree or from other easily accessed or topically appropriate menus. In an other example, the user can select clusters C.sub.1, C.sub.2, . . . C.sub.k listed on a particular menu M and choose to remove these clusters from the menu, replacing them on the menu with a single aggregate cluster M' containing all the target objects from clusters C.sub.1, C.sub.2, . . . C.sub.k. In this case, the immediate subclusters of new cluster M' are either taken to be clusters C.sub.1, C.sub.2, . . . C.sub.k themselves, or else, in a variation similar to the "scatter-gather" method, are automatically computed by clustering the set of all the subclusters of clusters C.sub.1, C.sub.2, . . . C.sub.k according to the similarity of the cluster profiles of these subclusters.

Detailed Description Text (350):
In one application, the browsing techniques described above may be applied to a domain where the target objects are purchasable goods. When shoppers look for goods to purchase over the Internet or other electronic media, it is typically necessary to display thousands or tens of thousands of products in a fashion that helps consumers find the items they are looking for. The current practice is to use hand-crafted menus and sub-menus in which similar items are grouped together. It is possible to use the automated clustering and browsing methods described above to more effectively group and present the items. Purchasable items can be hierarchically clustered using a plurality of different criteria. Useful attributes for a purchasable item include but are not limited to a textual description and predefined category labels (if available), the unit price of the item, and an associative attribute listing the users who have bought this item in the past. Also useful is an associative attribute indicating which other items are often bought on the same shopping "trip" as this item; items that are often bought on the same trip will be judged similar with respect to this attribute, so tend to be grouped together. Retailers may be interested in utilizing a similar technique for purposes of predicting both the nature and relative quantity of items which are likely to be popular to their particular clientele. This prediction may be made by using aggregate purchasing records as the search profile set from which a collection of target objects is recommended. Estimated customer demand which is indicative of (relative) inventory quantity for each target object item is determined by measuring the cluster variance of that item compared to another target object item (which is in stock).

Detailed Description Text (351):
As described above, hierarchically clustering the purchasable target objects results in a hierarchical menu system, in which the target objects or clusters of target objects that appear on each menu can be labeled by names or icons and displayed in a two-dimensional or three-dimensional menu in which similar items are displayed physically near each other or on the same graphically represented "shelf." As described above, this grouping occurs both at the level of specific items (such as standard size Ivory soap or large Breck shampoo) and at the level of classes of items (such as soaps and shampoos). When the user selects a class of items (for instance, by clicking on it), then the more specific level of detail is displayed. It is neither necessary nor desirable to limit each item to appearing in one group; customers are more likely to find an object if it is in multiple categories. Non-purchasable objects such as artwork, advertisements, and free samples may also be added to a display of purchasable objects, if they are associated with (liked by) substantially the same users as are the purchasable objects in the display.

Detailed Description Text (353):
The files associated with target objects are typically distributed across a large number of different servers S1-So and clients C1-Cn. Each file has been entered into the data storage medium at some server or client in any one of a number of ways, including, but not limited to: scanning, keyboard input, e-mail, FTP transmission. automatic synthesis from another file under the control of another computer program. While a system to enable users to efficiently locate target objects may store its hierarchical cluster tree on a single centralized machine, greater efficiency can be achieved if the storage of the hierarchical cluster tree is distributed across many machines in the network. Each cluster C, including single-member clusters (target objects), is digitally represented by a file F, which is multicast to a topical multicast tree MT(C1); here cluster C1 is either cluster C itself or some supercluster of cluster C. In this way, file F is stored at multiple servers, for redundancy. The file F that represents cluster C contains at least the following data:

Detailed Description Text (357):
The distributed hierarchical cluster tree can be created in a distributed fashion, that

is, with the participation of many processors. Indeed, in most applications it should be recreated from time to time, because as users interact with target objects, the associative attributes in the target profiles of the target objects change to reflect these interactions; the system's similarity measurements can therefore take these interactions into account when judging similarity, which allows a more perspicuous cluster tree to be built The key technique is the following procedure for merging n disjoint cluster trees, represented respectively by files F1 . . . Fn in distributed fashion as described above, into a combined cluster tree that contains all the target objects from all these trees. The files F1 . . . Fn are described above, except that the cluster labels are not included in the representation. The following steps are executed by a server S1, in response to a request message from another server S0, which request message includes pointers to the files F1 . . . Fn. 1. Retrieve files F1 . . . Fn. 2. Let L and M be empty lists. 3. For each file Fi from among F1 . . . Fn: 4. If file Fi contains pointers to subcluster files, add these pointers to list L. 5. If file Fi represents a single target object, add a pointer to file Fi to list L. 6. For each pointer X on list L, retrieve the file that pointer P points to and extract the cluster profile P(X) that this file stores. 7. Apply a clustering algorithm to group the pointers X on list L according to the distances between their respective cluster profiles P(X). 8. For each (nonempty) resulting group C of pointers: 9. If C contains only one pointer, add this pointer to list M; 10. otherwise, if C contains exactly the same subclusters pointers as does one of the files Fi from among F1 . . . Fn, then add a pointer to file Fi to list M; 11. otherwise: 12. Select an arbitrary server S2 on the network, for example by randomly selecting one of the pointers in group C and choosing the server it points to. 13. Send a request message to server S2 that includes the subcluster pointers in group C and requests server S2 to merge the corresponding subcluster trees. 14. Receive a response from server S2, containing a pointer to a file G that represents the merged tree. Add this pointer to list M. 15. For each file Fi from among F1 . . . Fn: 16. If list M does not include a pointer to file Fi, send a message to the server or servers storing Fi instructing them to delete file Fi. 17. Create and store a file F that represents a new cluster, whose subclusters pointers are exactly the subcluster pointers on list M. 18. Send a reply message to server S0, which reply message contains a pointer to file F and indicates that file F represents the merged cluster tree.

Detailed Description Text (358):
With the help of the above procedure, and the multicast tree MT full that includes all proxy servers in the network, the distributed hierarchical cluster tree for a particular domain of target objects is constructed by merging many local hierarchical cluster trees, as follows. 1. One server S (preferably one with good connectivity) is elected from the tree. 2. Server S sends itself a global request message that causes each proxy server in MT.sub.full (that is., each proxy server in the network) to ask its clients for files for the cluster tree. 3. The clients of each proxy server transmit to the proxy server any files that they maintain, which files represent target objects from the appropriate domain that should be added to the cluster tree. 4. Server S forms a request R1 that, upon receipt, will cause the recipient server S1 to take the following actions: (a) Build a hierarchical cluster tree of all the files stored on server S1 that are maintained by users in the user base of S1. These files correspond to target objects from the appropriate domain. This cluster tree is typically stored entirely on S1, but may in principle be stored in a distributed fashion. (b) Wait until all servers to which the server S1 has propagated request R have sent the recipient reply messages containing pointers to cluster trees. (c) Merge together the cluster tree created in step 5(a) and the cluster trees supplied in step 5(b), by sending any server (such as S1 itself) a message requesting such a merge, as described above. (d) Upon receiving a reply to the message sent in (c), which reply includes a pointer to a file representing the merged cluster tree, forward this reply to the sender of request R1, unless this is S1 itself 5. Server S sends itself a global request message that causes all servers in MT.sub.full to act on embedded request R1. 6. Server S receives a reply to the message it sent in 5(c). This reply includes a pointer to a file F that represents the completed hierarchical cluster tree. Server S multicasts file F to all proxy servers in MT.sub.full. Once the hierarchical cluster tree has been created as above, server S can send additional messages through the cluster tree, to arrange that multicast trees MT(C) are created for sufficiently large clusters C, and that each file F is multicast to the tree MT(C), where C is the smallest cluster containing file F.

Detailed Description Text (361):
Computer users frequently join other users for discussions on computer bulletin boards, newsgroups, mailing lists, and real-time chat sessions over the computer network, which may be typed (as with Internet Relay Chat (IRC)), spoken (as with Internet phone), or videoconferenced. These forums are herein termed "virtual communities." In current

practice, each virtual community has a specified topic, and users discover communities of interest by word of mouth or by examining a long list of communities (typically hundreds or thousands). The users then must decide for themselves which of thousands of messages they find interesting from among those posted to the selected virtual communities, that is, made publicly available to members of those communities. If they desire, they may also write additional messages and post them to the virtual communities of their choice. The existence of thousands of Internet bulletin boards (also termed newsgroups) and countless more Internet mailing lists and private bulletin board services (BBS's) demonstrates the very strong interest among members of the electronic community in forums for the discussion of ideas about almost any subject imaginable. Presently, virtual community creation proceeds in a haphazard form, usually instigated by a single individual who decides that a topic is worthy of discussion. There are protocols on the Internet for voting to determine whether a newsgroup should be created, but there is a large hierarchy of newsgroups (which begin with the prefix "alt.") that do not follow this protocol.

Detailed Description Text (434):
A separate multicast tree MT(V) is maintained for each virtual community V, by use of the following four procedures. 1. To construct or reconstruct this multicast tree, the core servers for virtual community V are taken to be those proxy servers that serve at least one pseudonymous member of virtual community V. Then the multicast tree MT(V) is established via steps 4-6 in the section "Multicast Tree Construction Procedure" above. 2. When a new user joins virtual community V, which is an existing virtual community, the user sends a message to the user's proxy server S. If user's proxy server S is not already a core server for V, then it is designated as a core server and is added to the multicast tree MT(V), as follows. If more than k servers have been added since the last time the multicast tree MT(V) was rebuilt, where k is a function of the number of core servers already in the tree, then the entire tree is simply rebuilt via steps 4-6 in the section "Multicast Tree Construction Procedure" above. Otherwise, server S retrieves its locally stored list of nearby core servers for V, and chooses a server S1. Server S sends a control message to S1, indicating that it would like to be added to the multicast tree MT(V). Upon receipt of this message, server S1 retrieves its locally stored subtree G1 of MT(V), and forms a new graph G from G1 by removing all degree-1 vertices other than S1 itself. Server S1 transmits graph G t o server S, which stores it as its locally stored subtree of MT(V). Finally, server S sends a message to itself and to all servers that are vertices of graph G, instructing these servers to modify their locally stored subtrees of MT(V) by adding S as a vertex and adding an edge between S1 and S. 3. When a user at a client q wishes to send a message F to virtual community V, client q embeds message F -in a request R instructing the recipient to store message F locally, for a limited time, for access by member s of virtual community V. Request R includes a credential proving that the user is a member of virtual community V or is otherwise entitled to post messages to virtual community V (for example is not "black marked" by that or other virtual community members). Client q then broadcasts request R to all core servers in the multicast tree MT(V), by means of a global request message transmitted to the user's proxy server as described above. The core servers satisfy request R, provided that they can verify the included credential. 4. In order to retrieve a particular message sent to virtual community V, a user U at client q initiates the steps described in section "Retrieving Files from a Multicast Tree," above. If user U does not want to retrieve a particular message, but rather wants to retrieve all new messages sent to virtual community V, then user U pseudonymously instructs its proxy server (which is a core server for V) to send it all messages that were multicast to MT(V) after a certain date. In either case, user U must provide a credential proving user U to be a member of virtual community V, or otherwise entitled to access messages on virtual community V.

Detailed Description Text (440):
Particularly within large organizations, it is advantageous to disseminate company (inside) news and information to those employees for whom the information is "valuable". Using the same basic profiling techniques (above). Virtual dialogues (either physical meetings or entirely virtual meetings, either e-mail or telephony based) may be automatically profiled on the fly and used for responsive indexing and notification of those users to whom the information is valuable (and to whom it is privy). As the content of such a dialogue may change with time, new users may be prompted to join while others may be prompted or alternatively (for confidentiality reasons) may be mandated to depart. Text summarization techniques may also be used to allow relevant users who missed the virtual meeting to have access to a synopsized version thereof. Document profiles of such meetings may also be organized into a hierarchical cluster tree using automatic cluster labeling or relevant terms within each cluster (Steve's reference hierarchical cluster menu trees from previous patent).

This technique is useful for intuitive browsing of large archives of this information). Digital credentials may be prescribed to each employee by superiors which indicate for him/her the specific information contexts (by clusters) which are mandatory, which are recommended, which are neutral, and which are inappropriate for the employee to either access or (for the mandatory credential) require also mandatory (real-time) attendance. A scheduling agent maybe used to organize meeting times in advance by contacting and informing the most relevant users as to the stated objectives of the meeting. This is done by coordinating available time slots to optimize the availability of the most number of user highest relevance users to the dialogue (the user may also indicate among his/her available time the level of convenience as well). As above suggested, in virtual work groups a virtual meeting's objective may be to solve a particular problem, and develop a strategy, plan or proposal the stated objective of which may be used to index a virtual group whose complement and skills provides an optimal solution thereto.

Detailed Description Text (442):
The above present methods may be used for retrieving documents by organizations to determine the relevance of internal correspondence (e-mail, fax, telephony and recorded physical dialogues) to the interests of the user as stated or exemplified. Thus all irrelevant correspondences are filtered out. Relevant ones may accordingly be clustered (labeled) and organized into a hierarchical cluster menu tree for industrial browsing as above described. For example, an employer may wish to "listen in" on certain types of correspondences with a particular client by a particular employee (via phone number and voice ID using Neural Net techniques) or about a particular topic. Again text summarization may aid the user in viewing large correspondences. In one approach fax, e-mail and telephone communications to and from each individual may also be monitored and advised similarly in order to enable the system to develop aggregate profiles for a given employee for both outgoing and incoming forms of each desired communication media which is used for purposes of routing.